Learning Converse with Personas

Yueran Liang
School of Computer and Communication Sciences, EPFL, Switzerland

***Abstract*—One of the main characteristics of human dialogues is the diversity of the responses: given an input message, there may exist multiple meaningful responses. The variety of responses might be attributed to different interpretations of the same utterance, which are led by divergent cultural backgrounds. In this project, two persona-based model will be presented including basic model and conditional variational autoencoder (CVAE) model. Both models aim to generate different responses according to different speakers with regard to the same utterance. For each model, it contains two sub-model which are Speaker Model to capture individual characteristics , and dyadic Speaker-Addressee Model aiming to discover properties of interactions between two interlocutors. Both models have been evaluated using perplexity and score derived by human evaluation. Comparison will be made between two models and find that persona-based CVAE model leads to a significant improvement on quality and perplexity compared with basic model.**

I. Introduction

In recent years, many researches in natural language processing area pay growing attention to developing high quality conversation models and constructing naturalistic conversation systems (Ritter et al., 2011[1]; Sordoni et al., 2015[2]; Vinyals and Le, 2015[3]; Li et al., 2016a [4]). To make the conversation more similar to human-to-human interaction in open-domain dialogue generation, persona-based model was proposed[5]. This model is able to generate diverse responses according to various speaking styles, by injecting speaker embedding and interaction embedding in Speaker Model and Speaker-Addressee Model respectively. It makes use of the sequence-to-sequence model (SEQ2SEQ)(Sutskever et al., 2014)[6] with encoder-decoder architecture. In addition, the model has also been utilized to construct personalized dialogue systems. Zhang et al.(2018)[7] proposed a model combining SEQ2SEQ and Memory Networks (Sukhbaatar et al., 2015)[8] with persona profile which is established exclusively and extracted from Facebook dataset with large amount of personal profile. Besides, an extensional personalizing dialogue agent was proposed by Madotto and Lin et al.(2019)[9] using model-agnostic meta-learning algorithm (Finn et al.,2017)[10]. Furthermore, adversarial learning framework (Olabiyi et al., 2019)[11] combining speaker embedding with the thriving generative adversarial network technique (Goodfellow et al., 2014)[12] also shows significant improvement on response diversity and capability to capture speaker characteristics.

It has been observed that some subtle differences in some words or mood are already capable to indicate the speaker's characteristic in speaking style. For example, passionate people tend to speak in intense tone so their responses should to be end with "!" instead of ".". In addition, some people may have mantra when they speak. Besides, Li et al. (2016b)[5] found that addressee also has significant influence on the answer that speaker responses to. Therefore, the aim of this project is to create diverse response that coincides in speakers' characteristics.

The model proposed in this project[1] is based on the work of Li et al. (2016b)[5] and its extension including Speaker Model and Speaker-Addressee Model. The contributions of this project include:

- Improve the consistency of response to context by utilizing Bahdanau attention(Bahdanau et al.,2016)[14] in the decoder to make it focus on the important part in the utterance.
- Modify the speaker embedding size proposed in the original paper to produce better results as the dataset in this project consists of only 14 main characters.
- Extend the original project by including conditional variational autoencoder (CVAE) model to enhance response diversity
- Compare two models by metrics including perplexity and human annotators' scores and find out that persona-based CVAE model shows significant improvement in both quality and perplexity.

II. Related Work

The structure of persona model consists of a backbone SEQ2SEQ model and two sub-models which inject specific embedding during the decoding process of the aforementioned backbone. SEQ2SEQ is an end-to-end technique which has been widely used in machine translation. From a general point of view, it consists of an encoder to encode the source sequence and a decoder to generate the target sequence for specific purposes such as machine translation. Some common choices of encoder and decoder include multilayered Long Short-Term Memory(LSTM)(Hochreiter and Schmidhuber, 1997)[13]. Two sub-models included in Persona are: a Speaker Model which models respondent alone and a Speaker-Addressee Model which is sensitive to interaction patterns. Specifically, in the Speaker Model, the speaker vector is injected to be personas so as to capture information of the speakers such as speaking

[1]https://github.com/yukixLLL/Persona_based-Conversation-Model

style. In Speaker-Addressee Model, the interactive vector is generated using the speaker vector which is to capture the interactive information. Both of speaker vector and interactive vector are built during training.

As for CVAE, it is an advanced model of VAE (Kingma and Welling, 2013[15]; Rezende et al., 2014[16]) which is very a popular framework in image generation. The latter technique is used to encode the input into a probability distribution z, e.g. $\mathcal{N}(0, \mathbf{I})$, instead of point encoding of autoencoder. In VAE, the decoder network then reconstructs the original input using samples from z. In contrast, CVAE generates diverse image conditioned on certain attributes e.g. generating different human faces given skin color (Yan et al., 2015[17]; Sohn et al., 2015[18]). Adapting CVAE to human conversation, Zhao et al. (2017)[19] succeeded in building a dialogue agent which are capable of generating diverse responses conditioned on dialogue contexts.

III. Models and Methods

A. Basic Persona-based SEQ2SEQ Model

In this section, the mechanism of the basic persona-based model is described in detail.

At the beginning, the encoder of model computes a hidden states vector h for each input sequence. Specifically, given a sequence of inputs $S = \{s_1, s_2, ..., s_n\}$, a vector h_t is obtained by the encoder at each instant t. The vector h_t, which is the hidden state of LSTM, combines K dimensional distinct word embedding e_t^S of an individual text unit s_t with the previous hidden state h_{t-1}.

The vector representation h_t for each time step t is generated by the following formula:

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ l_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} \cdot W \cdot \begin{bmatrix} h_{t-1} \\ e_t^S \end{bmatrix} \tag{1}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot l_t \tag{2}$$

$$h_t^S = o_t \cdot tanh(c_t) \tag{3}$$

where i_t, f_t, o_t denote an input gate, a memory gate and an output gate respectively. Meanwhile, c_t represents the cell state vector at time t, σ denotes the sigmoid function. $W \in \mathbb{R}^{4K \times 2K}$ is the parameter matrix which is composed of W^i, W^f, W^o and W^l. The LSTM mechanism is shown in the Figure 1. After that, the hidden states h^S are then used for response generation.

The response generation phase consists of the usage of the Speaker Model and the Speaker-Addressee Model.

As for the Speaker model, a speaker-level vector v_i is introduced as the representation of each individual speaker which encodes speaker-specific information. This vector will have an influence on the content and style of their response. Each speaker $i \in [1, N]$ has their distinct speaker embedding $v_i \in \mathbb{R}^{K' \times 1}$ where K' is not necessarily equal to the dimension of word embedding K. The speaker v_i is updated during the training process to better capture

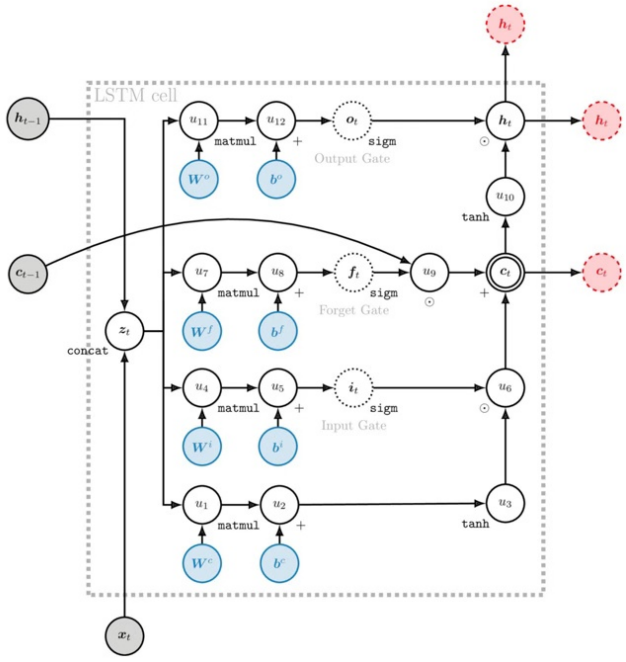

Fig. 1: Mechanism of LSTM where x_t refers to e_t^S and W^c refers to W^l

the speaker characteristics. For consistency, the speaker embedding v_i is shared across all the conversations that speaker i participates.

In order to let the decoder know the start and the end of the sentence, start-of-sentence symbol SOS and end-of-sentence symbol EOS are introduced. Moreover, let R denote the word sequence in response to S, where $R = \{SOS, r_1, r_2, ..., r_J, EOS\}$ and decoder will stop producing token when it meets EOS.

It has been unveiled by Bahdanau et al. (2016) that the original SEQ2SEQ model fails to consider the following two facts:

- Input parts may have different contextual importance
- Different input parts may not have the same importance to different output parts

To fix this issue, a localized context vector should be considered at each decoding step to enable the decoder to figure out which part of input is important in any instance of time. There are various ways to compute the context vector, the one computed by Bahdanau attention mechanism is:

$$score(h_t^R, \bar{h}_s^S) = v_a^T(tanh(W_1 \cdot h_t^R, W2 \cdot \bar{h}_s^S)) \tag{4}$$

$$\alpha_{ts} = \frac{exp(score(h_t^R, \bar{h}_s^S))}{\sum_{s'=1}^{S} exp(score(h_t^R, \bar{h}_{s'}^S))} \tag{5}$$

$$c_t = \sum_s \alpha_{ts} \bar{h}_s^S \tag{6}$$

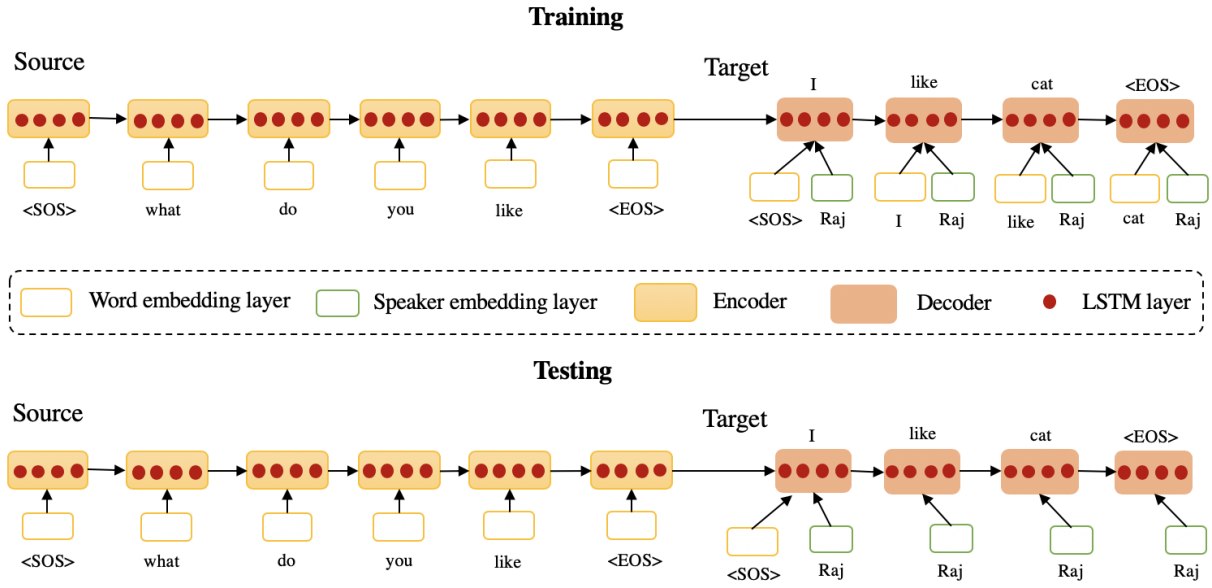

Fig. 2: Graphical description of basic persona-based SEQ2SEQ model with Speaker Model as example

where $\bar{h}_s^S$ corresponds to each input in a batch, v_a^T is a linear operator that maps the result of activation function to a integer value and parameter matrix $W1, W2 \in \mathbb{R}^{M \times M}$ in which M is the size of the hidden state.

In addition, the speaker embedding v_i is also injected at every time step so as to help predict personalized responses throughout the generation process. Hence, hidden state is obtained by combining the one produced at the previous step h_{t-1}, the word embedding e_t^R, the speaker embedding v_i and the context vector c_t:

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ l_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} \cdot W \cdot \begin{bmatrix} h_{t-1} \\ e_t^R \\ v_i \\ c_t \end{bmatrix} \tag{7}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot l_t \tag{8}$$

$$h_t^R = o_t \cdot tanh(c_t) \tag{9}$$

where the parameter matrix $W \in \mathbb{R}^{4K \times (2K + K' + M)}$. In this way, the model can capture information through the speaker embedding. Based on the personalities of different speakers, it will generate different responses to the same utterance.

As for the dyadic Speaker-Addressee model, the aim of this is to dig out the interactive information of two speaker i and j and predict how speaker i would respond to a message produced by speaker j. Similarly to the Speaker model, speaker-level representation are also needed for addressee and speaker which is denoted as v_i and v_j. Instead of directly combining them with hidden state, word embedding and context vector, two speaker embedding vectors should be linearly combined together first to obtain an interactive representation $V_{i,j} \in \mathbb{R}^{K' \times 1}$ using

$$V_{i,j} = tanh(W_i \cdot v_i + W_j \cdot v_j) \tag{10}$$

where $W_i, W_j \in \mathbb{R}^{K' \times K'}$. $V_{i,j}$ is then linearly incorporated into LSTM at each time step in the decoder:

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ l_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} \cdot W \cdot \begin{bmatrix} h_{t-1} \\ e_t^R \\ V_{i,j} \\ c_t \end{bmatrix} \tag{11}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot l_t \tag{12}$$

$$h_t^R = o_t \cdot tanh(c_t) \tag{13}$$

As $V_{i,j}$ depends on the speaker i and addressee j, the same speaker may thus respond differently to a utterance from different interlocutors.

In generation tasks of both models, the LSTM generates a distribution over outputs which defines the probability of each output $Y = \{y_1, y_2, ..., y_{n_Y}\}$ given the input $X = \{x_1, x_2, ..., x_{n_X}\}$ using a softmax function:

$$\begin{aligned} p(Y|X) &= \prod_{t=1}^{n_y} p(y_t|x_1, x_2, ..., x_t, y_1, y_2, ..., y_{t-1}) \\ &= \prod_{t=1}^{n_y} \frac{exp(f(h_{t-1}, e_{y_t}))}{\sum_{y'} exp(f(h_{t_1}, e_{y'}))} \end{aligned} \tag{14}$$

where $f(h_{t-1}, e_{y_t})$ denotes the activation function between h_{t-1} and e_{y_t}. Furthermore, beam search will be adopted for the word prediction in this experiment. The graphical description of the basic persona-based model in training and testing is shown in Figure 2.

B. Conditional Variational Autoencoder (CVAE)

Following III.1, the model is extended by using CVAE as the encoder.

In CVAE, each conversation can be represented via three random variables: the dialog c, the response x, and

the latent variable z which is sampled from the generated latent distribution. Then the conditional distribution can be defined as $p(x, z|c) = p(x|z, c)p(z|c)$. In this case, $p(z|c)$ and $p(x|z, c)$ can serve as prior network and response decoder respectively and both of them are approximated through the deep neural network (parameterized by θ). Then according to Zhao et al. (2017), the response generation process is defined as follows:

1) Sample a latent variable z from the prior network $p_\theta(z|c)$.
2) Generate x through the response decoder $p_\theta(x|z, c)$.

The graphical model of CVAE is shown in Figure 3.

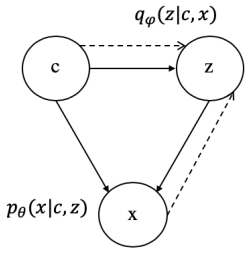

Fig. 3: Graphical model of CVAE

The goal of CVAE is to maximize the conditional log likelihood of x given c. Assume the latent distribution z follows multivariate Gaussian distribution with a diagonal covariance matrix and a recognition network is introduced to approximate the true posterior distribution $p(z|x, c)$. The variational lower bound proposed by Sohn and et al.(2015)[20] can be written as:

$$\begin{aligned}\mathcal{L}(\theta, \phi; x, c) &= -KL(q_\phi(z|x, c)\|p_\theta(z|c)) \\ &+ \mathbf{E}_{q_\phi(z|c,x)}[log\, p_\theta(x|z, c)] \\ &\leq log\, p(x|c)\end{aligned} \tag{15}$$

Moreover, since CVAE suffers from the vanishing latent variable problem (Bowman et al,. 2015)[21] which will make decoder fail to encode meaningful information in z, KL annealing, which is proposed by Bowman et al,. (2015), is introduced. This technique solves the problem by gradually increasing the weight of KL term from 0 to 1 while training. In addition, the paper also proposed a complementary technique, computing bag-of-word loss at the same time, to tackle this problem. The idea is to introduce an auxiliary loss that requires the decoder network to predict the bag-of-words in the response x. Response x is decomposed into two variable x_0 with word order and x_{bow} without order. Assume x_0 and x_{bow} are conditionally independent given z and c, then $p(x, z|c) = p(x_0|z, c)p(x_{bow}|z, c)p(z|c)$. In this way, the latent variable is forced to capture global information about the target response due to the conditional independence assumption.

Let $f = \mathbf{MLP}_b(z, x) \in \mathcal{R}^{|V|}$ where $|V|$ is the vocabulary size, and then yields:

$$log\, p(x_{bow}|z, c) = log \prod_{t=1}^{|x|} \frac{e^{f_{x_t}}}{\sum_j^{|V|} e^{f_j}} \tag{16}$$

where $|x|$ is the length of x and x_t is the word index of t_{th} word in x. Then the loss function of CVAE with bag-of-loss can be written as:

$$\begin{aligned}\mathcal{L}'(\theta, \phi; x, c) &= \mathcal{L}(\theta, \phi; x, c) \\ &+ \mathbf{E}_{q_\phi(z|c,x)}[log\, p_\theta(x_{bow}|z, c)]\end{aligned} \tag{17}$$

As for the model, Figure 4 illustrates the mechanism of the CVAE during training. Decoder in basic persona-based SEQ2SEQ model is reused. However, the encoder is changed to utilize bidirectional recurrent neural network (BRNN)(Schuster and Paliwal, 1997)[22] with a LSTM unit. This encoder encodes the source sequence into a vector by concatenating the last hidden states of the forward and backward BRNN $u_i = [\overrightarrow{h_i}, \overleftarrow{h_i}]$. While training, the original response is also encoded using the same encoder into a vector u_k. This vector will be fed in the recognition network afterwards. Since the latent distribution z is assumed to follow isotropic Gaussian distribution, the recognition network $q_\phi(z|x, c) \sim \mathcal{N}(\mu, \sigma^2\mathbf{I})$ and the prior network $p_\theta(z|c) \sim \mathcal{N}(\mu', \sigma'^2\mathbf{I})$, and then these two network can be constructed by:

$$\begin{bmatrix}\mu \\ log\,(\sigma^2)\end{bmatrix} = W_r \begin{bmatrix}x \\ c\end{bmatrix} + b_r \tag{18}$$

and

$$\begin{bmatrix}\mu' \\ log\,(\sigma'^2)\end{bmatrix} = \mathbf{MLP}_p(c) \tag{19}$$

respectively.

After that, reparameterization trick (Kingma and Welling, 2013)[15] is applied to acquire samples of z either from $\mathcal{N}(\mu, \sigma^2\mathbf{I})$ predicted by recognition network while training, or $\mathcal{N}(\mu', \sigma'^2\mathbf{I})$ predicted by prior network while testing. Then, decoder's initial state is generated by a generation network using $s_0 = W_i[z, c] + b_i$ and decoder starts to predict words sequentially (Shown in Figure 5).

C. Decoding and Response Generation

During decoding, the algorithm terminates when an EOS token is predicted. In order to make comparison between these two models, we apply the same approach on both models as follows. At each time step, the decoder generates N-best list using beam search with beam size $B = 3$. At each time step, the algorithm first examines all $B \times B$ possible next-word candidates, and add those finished sentences (end with EOS) which are of the top B candidates in the list. The unfinished hypothesis are preserved and moved to the next word position. This process continues until all B candidates are generated successfully or the algorithm has reached the maximum length of the sentence. The second response is picked as

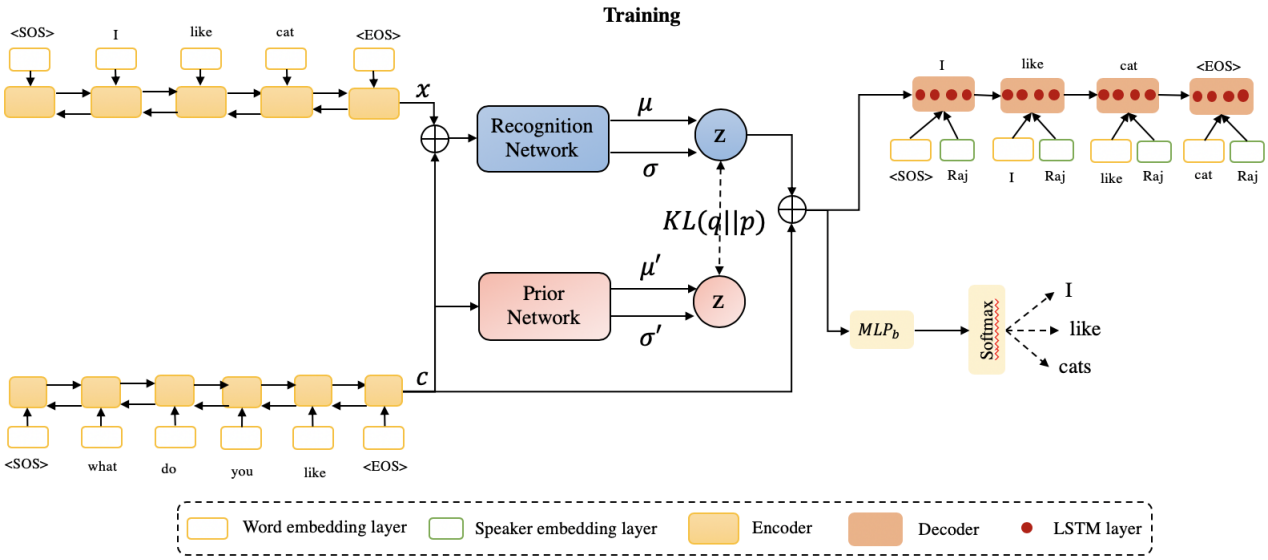

Fig. 4: Training process of CVAE with Speaker Model as example

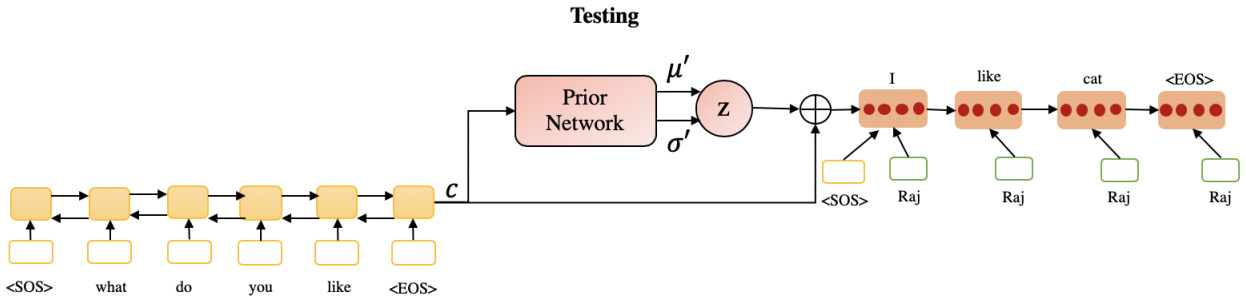

Fig. 5: Testing process of CVAE with Speaker Model as example

the final response. Although the first responses have higher probability, it tends to be short and generic and have less details corresponding to the utterance. However, the second ones are generally the extension of the first ones which are more diverse and contains more details about the context.

IV. Experiment Setup

A. Dataset

For both Speaker Model and Speaker-Addressee Model, scripts from the American television comedies *Friends*[2] and *The Big Bang Theory*[3] are chosen to be the dataset for evaluation. All the scripts are available from the internet[4][5].

To make the speaker individual characteristics be derived by the model easily, 13 main characters in the TV series [6] are considered to be the specific speakers who will have corresponding speaker vector representation in the models. We also group other characters in both TV series as the only one "Other" character. Adjective words and some other descriptive phrases shown in the speaker part are removed. For those utterances which are spoken by one more characters (e.g. Monica, Chandler, Phoebe, and Rachel), we assign these dialogues to the first main characters in the speaker list (e.g. Monica). After that, speaker ID from 1 to 14 are assigned to each speaker and then we collect the conversation among these 14 characters into the corpus. In addition, actions that are contained in the dialogues in the scripts are also removed. Dialogues in the same episode are then paired one by one. However, some pairs of dialogues do not contain an exact response to the utterance (e.g. just actions of characters). According to the mechanism of the model, these will create a large probability that model give a response with no words, so they are also removed from the corpus. Eventually, there are 112244 dialogue pairs in total.

[2]https://en.wikipedia.org/wiki/Friends
[3]https://en.wikipedia.org/wiki/The_ Big_Bang_Theory
[4]https://fangj.github.io/friends/ for *Friends*
[5]https://bigbangtrans.wordpress.com/ for *The Big Bang Theory*

[6]Joey, Rachel, Chandler, Monica, Ross, Phoebe, Leonard, Sheldon, Penny, Howard, Raj, Amy, Bernadette

The dialogue set needs to be pre-processed to adapt into the model. The pre-processing inclues:

1) For each dialogue sentence, create a space between a word and the punctuation (e.g. "he is a boy." ⇒ "he is a boy .") and replace everything with space except a-z, A-Z, ".", "?", "!" and ",".
2) Add the start-of-sentence and end-of-sentence symbols $< SOS >$ and $< EOS >$ in the beginning and the ending of the sentence respectively and then tokenize the sentence.
3) Create a vocabulary dictionary by offering each distinct word an index.
4) Truncate each sentence with max length 50 words, and pad those whose length do not reach the max with value 0 .
5) Convert word index sequence into Tensorflow tensor.

After shuffling, the corpus is split into training/validation/test sets, with training set of about 89795 turns and validation and test sets of 11224 and 11225 turns respectively.

B. Training

1) Basic Persona Model

Since the dataset is not very large, the hyperparameters are not as large as those mentioned in Li et al. (2016b). The Details of hyperparameters that have been tried in the basic persona SEQ2SEQ model are as follows:

- Use the default embedding layer in Tensorflow to implement word embedding with size 512.
- The number of LSTM layers in the encoder and decoder are both set to 4.
- The drop out rate is set to 0.2.
- The hidden size of encoder and decoder are the same and is chosen among the values 256, 400, 512, 600.
- Speaker embedding size is chosen between values 128 and 512.
- Use Adam as the optimizer and set the learning rate as 0.01.
- Batch size is set to 96.
- The number of epochs is set to 10.

We mainly tune the combination of hidden size and speaker embedding dimension in training to let the model generate response with better quality. After evaluated by the perplexity and the quality, the results show that:

- For Speaker Model, it is better to choose 400 size of hidden state and 512 size of speaker embedding;
- For Speaker-Addressee Model, it is better to choose 512 size of hidden state and 128 size of speaker embedding.

2) Persona CVAE

The Details of hyperparameters that have been tried in the persona CVAE model are as follows:

- Word embedding size is chosen between values 200 and 512.
- Speaker embedding is set to 128. (Because there is just tiny difference in perplexity in Speaker Model in the basic persona model, and the quality of response are similar no matter how large size is used.)
- The hidden size of encoder is chosen among values 300, 400, 512
- The hidden size of decoder is chosen between values 400 and 512
- The number of LSTM layers in the decoder is set to 4.
- The size of latent variable z is chosen between values 100 and 200
- The hidden size of prior network and recognition network is 2 times the latent size.
- The hidden size of bag-of-word fully connected neural network is set to 400
- The drop out rate of decoder is set to 0.2.
- Use Adam as the optimizer and set the learning rate as 0.01.
- Batch size is set to 64. (As the KL annealing matters, the batch size cannot be set to be so large.)
- The number of epochs is set to 10.

Furthermore, the steps of KL annealing is set to the half of the total steps which is #epochs $\times$ batch_size (Denote N). It means the weight of KL terms gradually increases in the first $0.5N$ steps and remains stable at 1 in the following.

In this case, we mainly tune the combination of the following hyperparameters: the hidden size of encoder and decoder, the word embedding dimension and the latent size. After perplexity and quality evaluation, the results that:

- For Speaker Model, it is better to choose 300 hidden size of encoder, 512 hidden size of decoder, 200 latent size and 512 word embedding dimension;
- For Speaker-Addressee Model, it is better to choose 400 hidden size of encoder, 400 hidden size of decoder, 200 latent size and 512 word embedding dimension.

V. Results

A. Quantitative Analysis

In the section, we will measure two main models by their perplexity which is the the exponentiation of the entropy loss. For persona CVAE model, other losses are also analysed.

There are actually two ways to compute overall perplexity of a model. One is to compute the exponential of the average entropy loss of the whole validation sets, which is applied in Li et al. (2016b). The other is to compute the perplexity of each sentence and then average the perplexity of the whole validation sets, which is the way used in Zhao et al. (2017). In order to make a comparison in not only coarse granularity but also fine granularity, both of these two ways are applied to compare between models.

The obtained coarse-grained perplexity of two model is reported in Table I. A significant decrease of perplex-

Perplexity	**Speaker Model**	**Speaker-Addressee Model**
Basic	25.7268	25.8111
CVAE	**19.5126** (-24.16%)	**19.0384** (-26.24%)

TABLE I: Perplexity Comparison in Coarse Granularity

ity over 20% can be observed for both Speaker Model and Speaker-Addressee Model when using CVAE model compared to the basic model. In particular, we can find the largest drop in perplexity in Speaker-Addressee Model (from 25.7268 to 19.0384) as we change the model from Basic to CVAE. Comparatively, Speaker Model experiences a smaller decrease in perplexity (from 25.7268 to 19.5126). Furthermore, with CVAE model, Speaker-Addressee Model shows better performance in perplexity than Speaker Model with 0.4742 difference. However, there is no obvious difference between these two sub-model when using basic persona model and the perplexity of Speaker-Addressee Model is even worse than the one of Speaker Model.

Perplexity	**Speaker Model**	**Speaker-Addressee Model**
Basic	115.39	125.67
CVAE	**107.56** (-6.79%)	**93.08** (-25.93%)

TABLE II: Perplexity Comparison in Fine Granularity

The obtained fine-grained perplexity of two model is reported in Table II. There is still a decrease in perplexity changing the model from basic to CVAE in either Speaker Model or Speaker-Addressee Model. The drop is still evident in the Speaker-Addressee Model with 25.93% even in this case. In contrast, perplexity in Speaker Model descends not so obviously which drops just from 115.39 to 107.56. Moreover, similarly to the coarse case, Speaker-Addressee Model has higher perplexity in basic model but lower in CVAE model where it even reaches a value under 100. In addition, the difference between two sub-model in CVAE with value 14.48 is still larger than that in basic model. This may indicate that using SAM in CVAE would generate better responses compared to using SM in CVAE.

Loss	**Speaker Model**	**Speaker-Addressee Model**
ELBO	**7.1597**	7.8194
Entropy Loss	2.9711	**2.9465**
KL Loss	**4.1887**	4.8729
BOW Loss	77.4899	**76.3119**

TABLE III: Loss of CVAE Comparison

Every type of loss of CVAE model is reported in Table III. From the table, it shows that although Speaker-Addressee Model reaches lower perplexity, its variational lower bound (ELBO) is still higher than Speaker Model. It means the probability that the model can generate a response matching the original response is actually lower. In addition, the KL loss of Speaker-Addressee Model is also larger than that of Speaker Model, which may indicate the prior network of Speaker Address Model does not match the recognition network as well as Speaker Model. However, Speaker-Addressee Model still gets a lower bag-of-word (BOW) loss and entropy loss than Speaker Model. It indicates that this model has a better performance on generating response with similar elements to the original response (As referred to III.2, x_{bow} does not have word order), which also shows the diversity of this model.

B. Qualitative Analysis

1) Different Responses by Different Speaker

In this section, Speaker Model and Speaker-Addressee Model are evaluated separately. For Speaker Model, 6 main characters in each TV series are chosen to generate responses to the same source message. Results will be evaluated whether different speakers produce different responses and the quality of the responses. For Speaker-Addressee Model, one main character is chosen randomly to be the speaker and then 3 other main characters are chosen to be the addressees who speak the source message. In this case, results will be evaluated if the speaker responds differently to different addressee and quality as well. Because of different background of these TV series, results are shown separately according to the series.

Source message: Oh. so, how many children do you think we should have? I'm sorry , that was a little abrupt.	
Basic Persona	
Leonard:	I'm sorry .
Sheldon:	I don't know . I don't think so .
Penny:	Oh , I'm sorry .
Howard:	I don't know . I'm sorry .
Raj:	I don't know . I'm sorry .
Amy:	I don't know . I'm sorry .
CVAE	
Leonard:	Okay , let's just talk about the rest of my life.
Sheldon:	I'm sorry . what are you doing?
Penny:	Yeah , well , I've been thinking about it.
Howard:	What?
Raj:	Oh , that's so sweet.
Amy:	You know , I was wondering if you were a kid.

TABLE IV: Speaker Model in the Big Bang Theory

Table IV shows the resulting responses generated with characters in *The Big Bang Theory*. Comparing between two models, we can observe that responses generated by CVAE model have better performance in diversity. Each character can give their own distinct response to the message. However, although the basic persona model generates response with modal like "Oh", which can represent the speaking style of the speaker, most of the answers seem to "feel sorry" most of the time. Furthermore, for the source message, there are two ways to respond to the message. One of the ways is to answer the first question and the other is to respond with the attitude towards "a little abrupt". Different responses by different ways show

different attitudes of the speakers, which is also a kind of characteristics. As shown in table, most characters pay attention to the latter sentence and feel embarrassed or offended. However, Raj, who is the only one bachelor in these characters, seems to be willing to put more emphasis on the question and feel it "sweet".

Source message: You've spoiled everything ! It's like a nightmare ! My friends and family are out there! How can I face them ? ! How can you do this to me?!	
Basic Persona	
Joey:	What?
Rachel:	What?
Chandler:	What?
Monica:	What?
Ross:	What?
Phoebe:	What?
CVAE	
Joey:	Well , I don't know .
Rachel:	Okay , you're a big deal!
Chandler:	What?
Monica:	Well , I mean , I mean , it's not like I've been thinking about it .
Ross:	What?
Phoebe:	Oh , I'm sorry .

TABLE V: Speaker Model in Friends

Table V shows the responses produced by characters in *Friends*. In this case, basic persona model produces very unsatisfactory responses. As we observe, even though the characters' personalities are different, it has produced the same answer for all of them. Furthermore, this answer does not make much sense. However, CVAE model still gives diverse responses although there are some weird answers like "What?" said by Chandler and Ross. Characters also show different attitudes to the message. Phoebe seems to feel very sorry about the situation indicated in the message while Joey and Monica seem to feel helpless in it.

To sum up, CVAE model have better performance on both diversity and quality to the response than basic persona model in Speaker Model. The basic model tends to generate meaningless and generic answer to the message regardless of speakers and thus shows no characteristic. Comparatively, CVAE tends to generate different kinds of answers and shows stronger characteristics of the speakers.

Source message: i m being awful ? you re the one who went out of your way to hurt me .	
Basic Persona	Addressee: Sheldon
Leonard:	I'm sorry.
CVAE	Addressee: Sheldon
Leonard:	Oh , you're right.
Basic Persona	Addressee: Penny
Leonard:	I'm sorry , I'm sorry .
CVAE	Addressee: Penny
Leonard:	I'm sorry.
Basic Persona	Addressee: Raj
Leonard:	I'm sorry.
CVAE	Addressee: Raj
Leonard:	Oh, that's funny.

TABLE VI: Speaker-Addressee Model in the Big Bang Theory

Table VI shows the utterance produced by Leonard in response to the source message spoken by 3 other characters in *The Big Bang theory*. The generated responses with the basic model seem to make sense in some way and Leonard even repeats "I'm sorry" twice to his wife Penny to show his great apology. It shows the model seems to have captured the interactive patterns successfully. However, CVAE seems to give more reasonable response. When the addressee is Sheldon, Leonard's response is actually very reasonable as he really thinks Sheldon is selfish. And his response to Penny is also reasonable similarly to the basic model. Additionally, since Raj is Leonard's friend, the response may also be reasonable to some extent.

Source message: Well , no , no , wait , wait , wait. All right , I gotta go. Just listen. Promise me, that you will wait a minute before you call her.	
Basic Persona	Addressee: Phoebe
Joey:	Oh , I'm sorry.
CVAE	Addressee: Phoebe
Joey:	Okay.
Basic Persona	Addressee: Chandler
Joey:	I'm sorry.
CVAE	Addressee: Chandler
Joey:	Why?
Basic Persona	Addressee: Monica
Joey:	I'm sorry.
CVAE	Addressee: Monica
Joey:	All right , let's see what's going on .

TABLE VII: Speaker-Addressee Model in Friends

Table VII shows the utterance produced by Joey in response to the source message spoken by 3 other characters in *Friends*. The performance of basic persona model seems to be a disaster. Similarly to what is mentioned in the Speaker Model, the generated responses are quite weird and somehow meaningless and shows little characteristic of the speaker. By contrast, the responses produced by CVAE model still show better performance and the first two response are actually very close to the ground truth answer "Okay. Why?". Moreover, CVAE can generate response according to the addressee although it is short. As for the last response to Monica, it is somehow strange and seems not to be a reasonable answer in this situation.

In summary, CVAE model also outperforms basic persona model in Speaker-Addressee mode. It can still generate diverse answers corresponding to different addressee with more rationality while the basic model tends to produce some weird and meaningless answers.

2) Human Evaluation

The human evaluation is conducted of outputs from both basic persona model and CVAE model. Fifty source messages are randomly chosen and fed to these two models having each two sub-models. Each model generates a response for each message, which makes a total of 200 responses. Then, we match each response of each model to the corresponding source message. We therefore have 50 groups of response messages, and each group contains the

4 responses generated by the different models. To evaluate the quality of these response messages, we send them to three annotators and ask them to score these messages. All the models are judged on a 5-point zero-sum scale as in Li et al. (2016b). The annotators will grade -2 if they find the response's quality poor; -1 if they find it weird; 0 if fair or hard to judge; 1 if the response message makes sense to them; and 2 if they think that the response is character-like. The results of human evaluation is shown in the Table VIII.

Score	Basic Speaker Model	Basic Speaker-Addressee Model	CVAE Speaker Model	CVAE Speaker-Addressee Model
Poor(-2)	75	108	43	23
Weird(-1)	24	22	45	43
Fair(0)	23	8	17	21
Make Sense(1)	24	11	36	48
Character-like(2)	4	1	9	15
Acceptance %	2.7	0.67	30	42

TABLE VIII: Results of Human Evaluation

As is shown in the table, CVAE model gains more positive scores than basic persona model in general. Either Speaker Model or Speaker-Addressee Model has a significant improvement according to the gained scores of the character-like one. In particular, the Speaker-Addressee Model of CVAE obtains the most character-like scores among these 4 models, which is consistent to the result of perplexity. Furthermore, we consider the responses are acceptable if they make sense or are character-like. The acceptance rates of both sub-model of CVAE are impressively higher than those of basic persona model with 30% for Speaker Model and 42% for Speaker-Addressee Model. In contrast, basic persona model has very awful results on score. Especially for Speaker-Addressee Model, it gets 108 "poor" scores and only 0.67% acceptance rate.

VI. Conclusion

In conclusion, CVAE model shows better performance than basic persona model in either Speaker Model or Speaker-Addressee Model. Both of these two sub-model have better perplexity values and better quality of the generated sentences. In addition, CVAE model can generate diverse responses according to different speakers or different addressees, which can give more reasonable answers to message and more possibly show characteristics of speakers. By contrast, basic persona model shows poor performance in not only perplexity but also quality. Most importantly, it tends to generate very dull, generic and sometimes meaningless responses and shows no speaker individual characteristics.

Moreover, response consistency is actually a very important aspect that needs to be paid attention to in the personalized model. Apart from the speaking style, background information should also be captured by the model. For example, if the speaker is asked about the following two question "What do you do?" and "What is your job", speaker should give similar answers in response. Furthermore, the generation capacity should also be expected because the speaker embedding can refer to similar information if they are close in embedding dimension. This is very important since the training data does not contain explicit information about every attribute of each user (e.g. gender, age,country of residence). For instance, Rob lives in England with high probability if he often get along with people also in England. However, in order to get these kind of outcomes, model should be fed with a dataset with large amount of personal information like Twitter and Facebook. But there is no right for us to access to these kind of dataset.

Although the basic persona model seems like a disaster in this project, it can be improved by methods mentioned in Li et al.(2016b). As mentioned in this paper, they first pre-train the SEQ2SEQ model and word embedding on a very large conversation dataset OpenSubtitles (Tiedemann,2009)[23], which does not contain specific speakers. However, the OpenSubtitles is extremely large that it will consume a large amount of time in pre-training process. While decoding, they generate 200-best response list and then re-ranking them by using MERT algorithm[24] whereas Zhao et al. (2017) just use the most possible words. In order to make a reasonable comparison, same beam search is applied to two models. In addition, according to the result shown in this paper, Speaker-Addressee Model should have better performance than the Speaker Model. Moreover, it can generate response with the addressee's name in the end of the answers with high probability. This shows obvious interactive information in the conversation, since the response can detect the name of the addressee, which is a strong characteristic of the addressee. Although the CVAE model seems to give better response, it is more possible to generate a response that just makes sense instead of being really character-like. Therefore, from our point of view, these kind of methods may also lead an improvement of the CVAE model.

References

[1] Alan Ritter, Colin Cherry, and William B Dolan. (2011). Data-driven Response Generation in Social Media. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 583593.

[2] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. (2015). Building End-to-end Dialogue Systems Using Generative Hierarchical Neural Network Models. In *Proc. of AAAI.*

[3] Oriol Vinyals and Quoc Le. (2015). A Neural Conversational Model. In *Proc. of ICML Deep Learning Workshop.*

[4] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. (2016a). A Diversity-promoting Objective Function for Neural Conversation Models. In *Proc. of NAACL-HLT.*

[5] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. (2016b). A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155.*

[6] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. (2014). Sequence to Sequence Learning With Neural Networks. In *Advances in neural information processing systems (NIPS)*, pages 3104–3112.

[7] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, Jason Weston. (2018). Personalizing Dialogue Agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*.

[8] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. (2015). End-to-end memory networks. In*Advances in Neural Information Processing Systems*, pages 2440–2448.

[9] Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, Pascale Fung (2019). Personalizing Dialogue Agents via Meta-Learning. In*Association for Computational Linguistics*, pages 5454–5459.

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In*Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org.

[11] Oluwatobi Olabiyi, Anish Khazane, Alan Salimov, Erik T. Mueller (2019). An Adversarial Learning Framework For A Persona-Based Multi-Turn Dialogue Model *arXiv preprint arXiv:1905.01992*.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. (2014). Generative Adversarial Networks In *Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014)*, pages 2672–2680.

[13] Sepp Hochreiter and Jürgen Schmidhuber. (1997). Long Short-term Memory. *Neural computation*, 9(8):1735–1780.

[14] Dzmitry Bahdanau, KyungHyun Cho and Yoshua Bengio. (May 19, 2016). Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv preprint arXiv:1409.0473*.

[15] Diederik P Kingma and Max Welling. (May 1, 2014). Auto-encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*.

[16] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. (May 30, 2014). Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *arXiv preprint arXiv:1401.4082*.

[17] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. (2015). Attribute2image: Conditional image generation from visual attributes. *arXiv preprint arXiv:1512.00570*.

[18] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. (2015). Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491.

[19] Tiancheng Zhao, Ran Zhao, Maxine Eskenazi (2017). Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders *arXiv preprint arXiv:1703.10960*.

[20] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. (2015). Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems.* pages 3483–3491.

[21] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. (2015). Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

[22] Mike Schuster and Kuldip K Paliwal. (1997). Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

[23] Jörg Tiedemann. (2009). News from OPUS – a collection of multilingual parallel corpora with tools and interfaces. In *Recent advances in natural language processing*, volume 5, pages 237–248.

[24] Franz Josef Och. (2003). Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July. Association for Computational Linguistics.